

Проектирование информационной системы для риэлтерской компании

Волков Виталий Александрович

Мордовский государственный университет им. Н.П.Огарева

студент

Аннотация

В данной статье рассмотрено объектно-ориентированное проектирование информационной системы для обеспечения бизнес-процессов, связанных с обменом квартир и поиском вариантов обмена.

Ключевые слова: Маклер, ИС, UML, проектирование системы, актор, прецедент, диаграмма последовательности, класс, база данных

Design of an information system for the realtor company

Volkov Vitaliy Alexandrovich

Ogarev Mordovian State University

student

Abstract

In this article object-oriented design of an information system for support of the business processes connected to exchange of apartments and search of options of exchange is considered.

Keywords: Broker, IC, UML, system design, actor, precedent, chart of the sequence, class, database

Сложность риэлтерского бизнеса состоит в том, что существует два предмета продажи, такие как услуга и объект. Продажа услуги определяется типичными методами и проблемами сферы услуг. Объект недвижимости – это материальный объект, с уникальными характеристиками. Продвижение того и другого отличаются и имеют свои законы.

Информация, как составляющая и того и другого процесса, имеет одинаковые характеристики, но подход к технологии обработки и эффективного использования другой. К тому же, особенности развития рынков и менталитета потребителей, усложняют процесс внедрения информационных технологий в этой отрасли. Поэтому исследования проектирования и разработки информационных системы для обеспечения деятельности риэлтерских компаний являются актуальными.

Информационная система «Маклер» для учета заявок на обмен квартир и поиска вариантов обмена должна предоставить пользователю возможность осуществлять просмотр данных по заявкам на сдачу и аренду жилья (количество комнат, площадь, этаж, район). Система должна обеспечивать

выполнение таких функций как: ввод заявки на обмен; поиск в картотеке подходящего варианта; вывод всей картотеки.

Данная система должна позволять оператору вводить всю информацию которая необходима. При работе с системой имеется возможность использования информации хранящейся в базе данных. Также предусмотрена возможность просмотра, редактирования и печати документов из базы данных. Все данные, хранящиеся на сервере базы данных, доступны по локальной сети предприятия, для любого компьютера, где установлена клиентская часть системы (рис. 1).

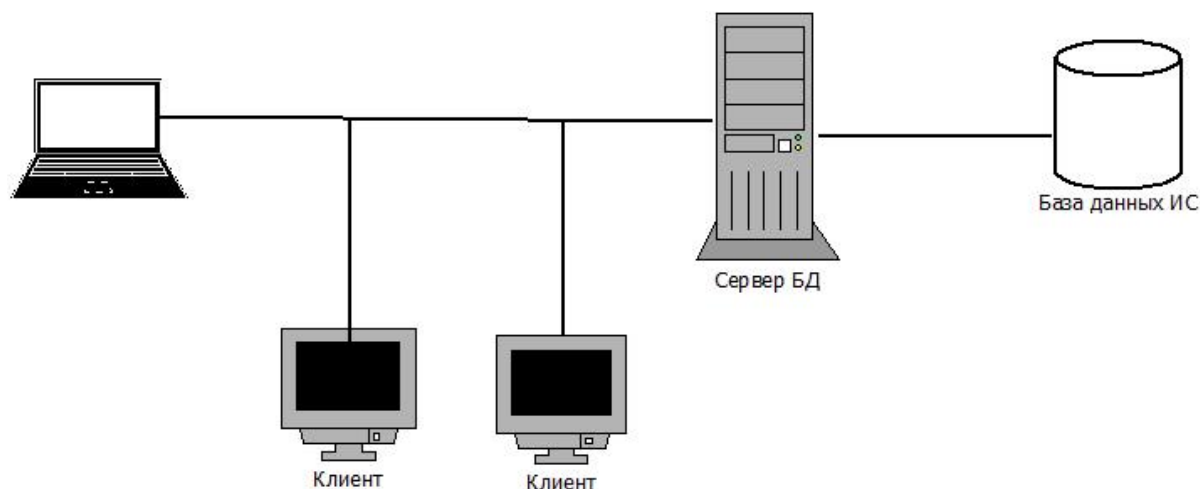


Рисунок 1 – Общая картина решения

Бизнес-цели для создания продукта приведены в таблице 1.

Таблица 1 – Перечень бизнес-целей

Идентификатор	Бизнес-цель
ВА-1	Информационное обеспечение работы риэлтерской компании

Задачи для которых предназначена система приведены в таблице 2.

Таблица 2 – Перечень бизнес-задач

Идентификатор	Бизнес-задача
ВТ-1	Ведение справочника клиентов
ВТ-2	Ведение реестра квартир на обмен
ВТ-3	Учет и контроль сделок по обмену

В системе были выделены такие акторы как:

Маклер - оператор ввода данных, который отвечает за поддержание в актуальном состоянии базы данных системы.

Руководитель организации, который имеет возможность просмотра и анализа отчетов по сделкам.

Клиент, который имеет возможность просмотра и поиска искомых вариантов обмена.

На основе анализа требований к информационной системе была построена диаграмма вариантов использования системы представленная на рисунке 2 и определен перечень акторов и вариантов использования системы представленные на таблицах 3 и 4.

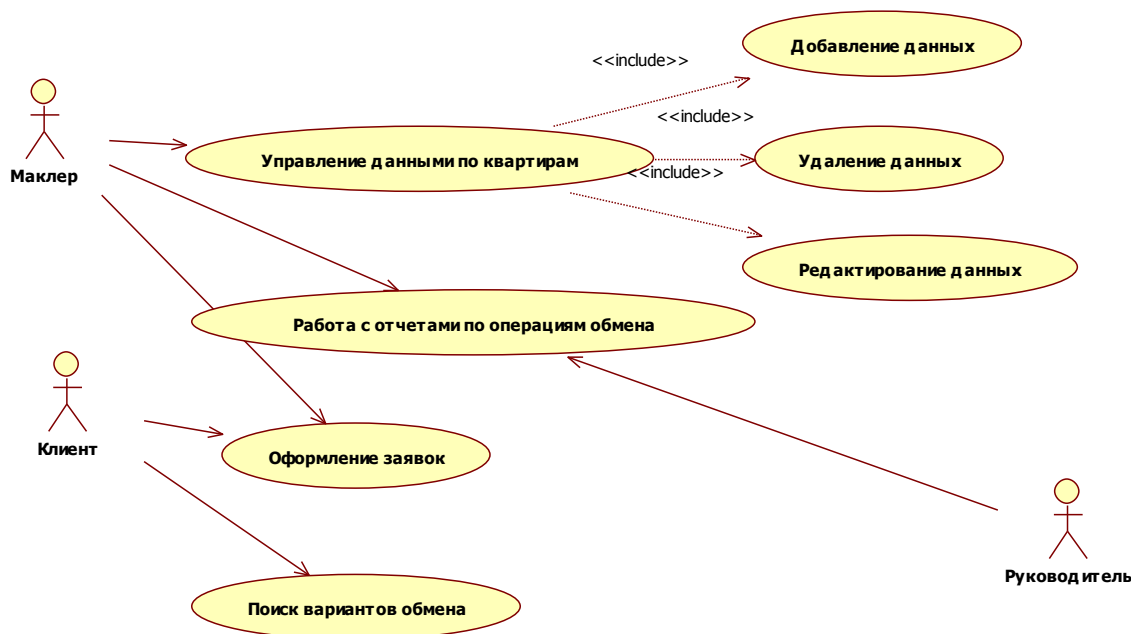


Рисунок 2 – Диаграмма вариантов использования системы

Таблица 3. – Перечень акторов

Идентификатор	Актер
АС-1	Маклер
АС-2	Клиент
АС-3	Руководитель

Таблица 4 – Перечень вариантов использования

Действующее лицо	Идентификатор	Описание
АС-1	UC-1	Формировать данные по квартире
АС-1	UC-2	Формировать отчеты по операциям
АС-1	UC-3	Оформление заявок на обмен
АС-2	UC-3	Оформление заявок на обмен
АС-2	UC-4	Поиск вариантов обмена
АС-3	UC-2	Формировать отчеты по операциям

Далее построим диаграмму последовательности. Для неё главным моментом является динамика взаимодействия объектов во времени. При этом она как бы имеет два измерения. Одно является линией жизни отдельного

объекта и представлено слева направо, а другое - сверху вниз и является вертикальной временной осью. Взаимодействие между объектами реализуется путём обмена сообщениями объектов друг с другом. Сообщения появляются в таком порядке, в котором они представлены на диаграмме, т.е. сверху вниз. Диаграмма последовательностей для варианта использования UC3 представлена на рисунке 3.

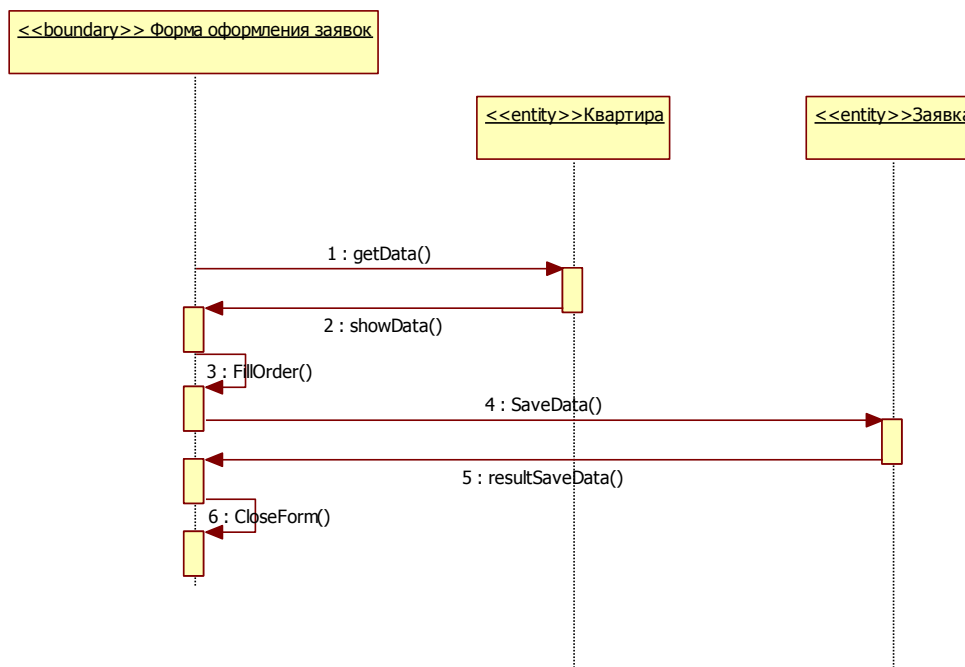


Рисунок 3 – Диаграмма последовательности

Объекты - это люди либо предметы, которые описаны в сценариях использования системы. На их основе создаются сервисы, отношения и атрибуты. Объекты проектируемой системы представлены в таблице 5.

Таблица 5 – Объекты проектируемой системы

Объект	Вариант использования	Описание
Маклер	UC-1, UC-2, UC-3,	Маклер формирует данные по квартирам и заявкам
Квартиры	UC-1	Данные для по квартирам
Заявки	UC-3	Отчет по операциям обмена
Отчеты	UC-2	Заявки клиентов на обмен
Клиенты	UC-3, UC-4	Клиент выполняет поиск вариантов обмена и подачу заявок

Построим диаграмму классов нашей системы. Она используется для построения логической модели системы и включает в себя классы-сущности, их атрибуты и взаимосвязей между ними.

В языке UML класс служит для обозначения множества объектов, которые обладают одинаковой структурой, отношениями и поведением с объектами из других классов.

На основе анализа предметной области, были выявлены следующие классы – сущности нашей системы: Клиент, Маклер, Квартира, Заявка. Описание данных классов и их атрибутов представлены в таблице 6.

Таблица 6 – Объекты предметной области и концептуальные классы

Объект	Класс-сущность	Атрибуты
Клиент	Клиент	Фамилия Имя Телефон
Маклер	Маклер	Фамилия Имя Телефон
Квартира	Квартира	Количество комнат Этаж Район Площадь
Заявка	Заявка	Дата заявки Квартира Клиент

Модель предметной области представлена в виде диаграммы классов на рисунке 4.

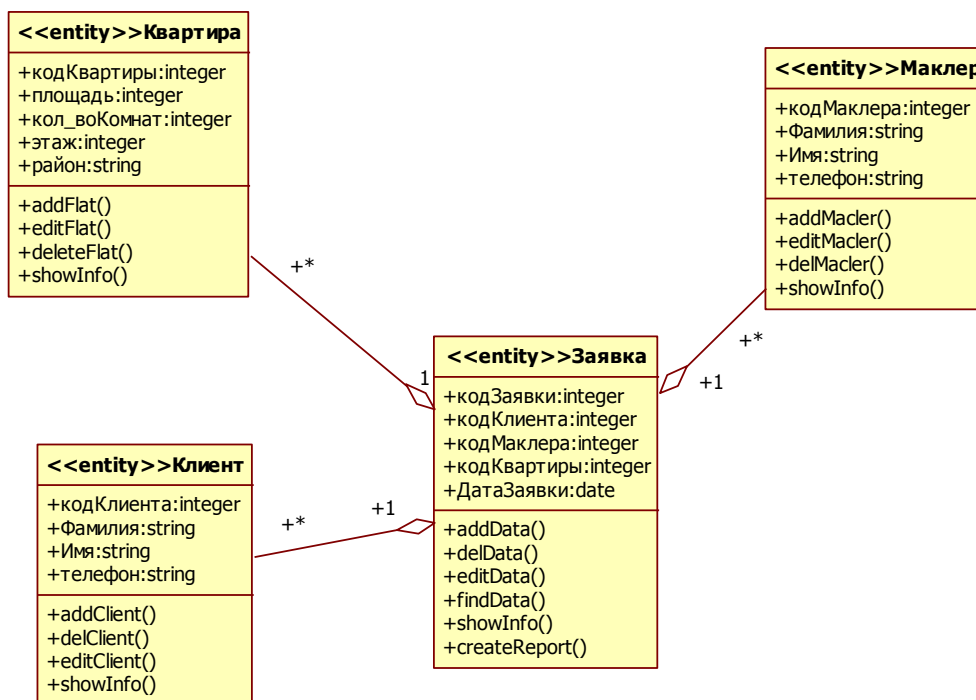


Рисунок 4 – Диаграмма классов

Взаимодействие объектов в системе происходит путём приема и передачи сообщений объектами-клиентами и обработки этих сообщений объектами-серверами. Чтобы отразить данную последовательность передачи сообщений между объектами, построим диаграмму кооперации, которая используется для построения модели динамического взаимодействия объектов. Она отражает структурную организацию объектов, принимающих или отправляющих сообщения. Данная диаграмма представлена на рис. 5.

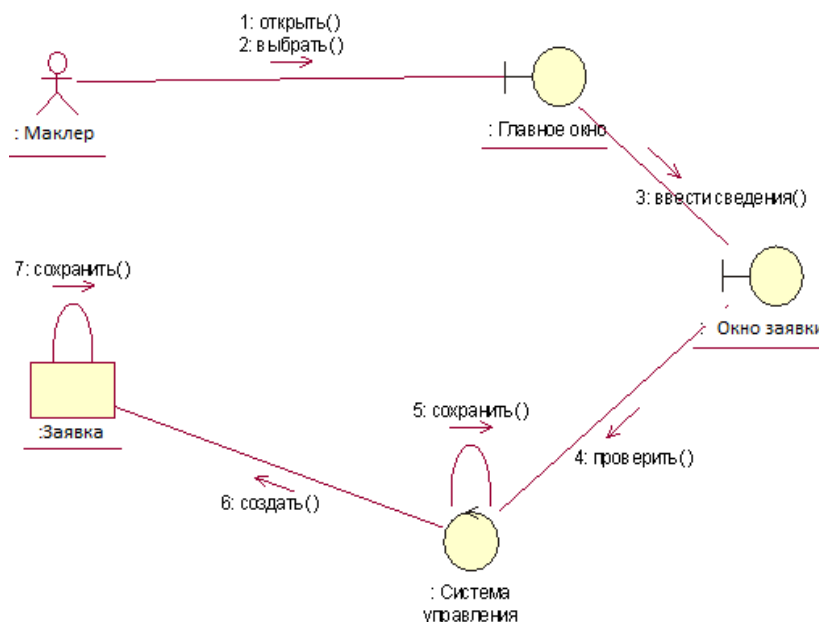


Рисунок 5 – Диаграмма кооперации

Для реализации проектируемой системы будет использована клиент-серверная архитектура. Такая архитектура позволит хранить все данные на сервере базы данных. Для этого построим диаграмму компонентов, представленную на рисунке 6.

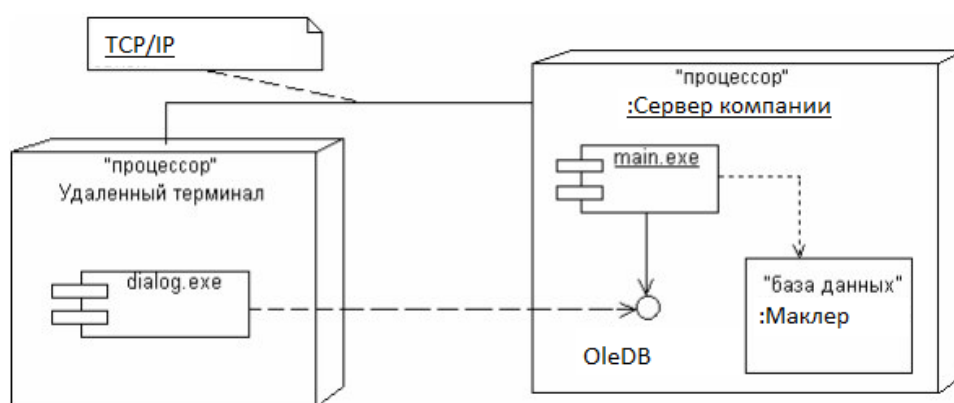


Рисунок 6 – Диаграмма компонентов

Чтобы спроектировать реляционную базу данных необходимо выделить определенную совокупность таблиц нашей системы, которые будут содержать необходимую информацию. А также установить связи между

данными таблицами. Логическая модель данных для системы представлена на рисунке 7.

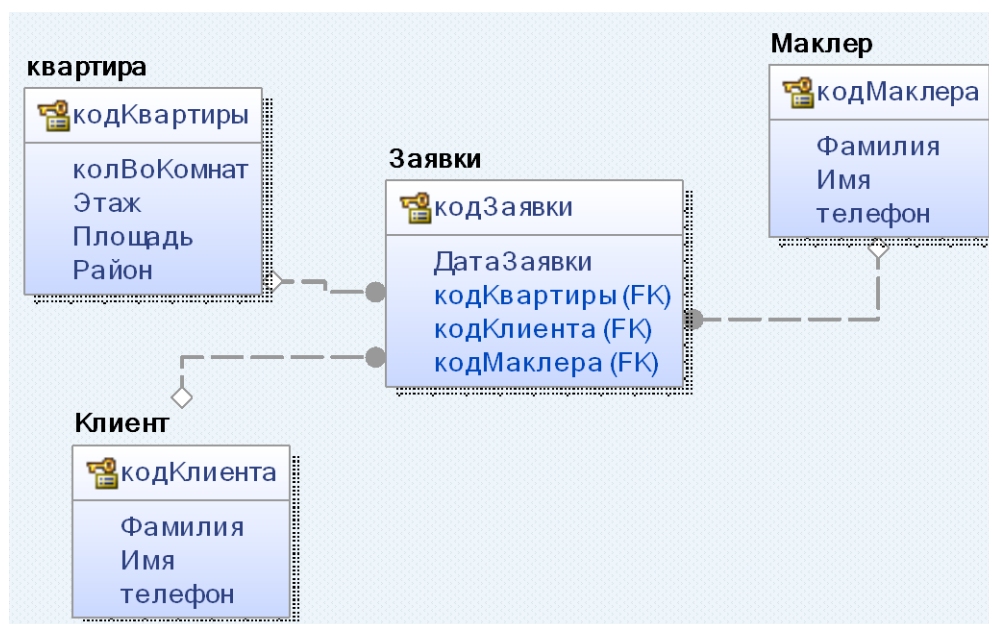


Рисунок 7 – Логическая модель данных

Спроектированная нами система позволит значительно сократить затраты финансовых, временных и материальных ресурсов риэлтерской компании. Также позволит снизить нагрузку на менеджеров данных компаний.

Таким образом, данная система будет полезна для применения при решении разнообразных задач программирования специалистами конкретной прикладной области. Благодаря заложенной в архитектуре концепции расширяемости, сторонние разработчики могут создавать свои модули для увеличения круга применимости разработанного программного продукта.

Библиографический список

1. Фаулер М. UML. Краткое руководство по стандартному языку объектного моделирования. СПб: Символ-Плюс, 2011. 192 с.
2. Боггс У., Боггс М. UML и RationalRose. М.: Лори, 2008. 600 с.
3. Волков В.А. Работа на C++ с MySQL // Постулат. 2016. № 12(14). С.49.
4. Волков В.А. Исторические шифры на примере языка C // Постулат. 2017. № 4 (18). С.61.
5. Волков В.А. Работа на Assembler в Microsoft Visual Studio 2015 // Постулат. 2017. № 5.